A least squares based finite volume method for the Cahn-Hilliard and Cahn-Hilliard-reaction equations

S. Dargaville, T.W. Farrell

September 10, 2012

Abstract

A vertex-centred finite volume method (FVM) for the Cahn-Hilliard (CH) and recently proposed [1, 2] Cahn-Hilliard-reaction (CHR) equations is presented. Information at control volume faces is computed using a high-order least-squares approach based on Taylor series approximations. This least-squares problem explicitly includes the variational boundary condition (VBC) that ensures that the discrete equations satisfy all of the boundary conditions. We use this approach to solve the CH and CHR equations in one and two dimensions and show that our scheme satisfies the VBC to at least second order. For the CH equation we show evidence of conservative, gradient stable solutions, however for the CHR equation, strict gradient-stability is more challenging to achieve.

1 Introduction

The Cahn-Hilliard (CH) equation was first proposed to model the quenching of binary alloys [3], but has found uses modelling many types of phase separation behaviour. Numerically, the CH equation is difficult to solve accurately, especially in two or three dimensions; it is very stiff and possesses multiple time and space scales. Our interest in CH problems is motivated by semi-conductor intercalation processes, such as those that occur in $LiFePO_4$ batteries and other electrochemically active porous media. $LiFePO_4$ is a promising battery material [4] that undergoes phase-separation during charge and discharge to form highly and lowly lithiated phases [4, 5]. Recently, Singh et al. [1] used the CH equation to model phase separation in this material. Previously, a Stefan problem [6] (which produces isotropic "shrinking-core" behaviour on a spherical domain) has been used to model this phenomenon [7, 8]. However, such a mechanism does not reflect the experimentally observed anistropy of Li^+ diffusion in $LiFePO_4$ [9] and it is this acute anistropy that motivates the CH approach of Singh et al. [1].

Traditionally, a no-flux boundary condition (BC) is applied to the CH equation. Singh et al. [1] however, used an anisotropic, two-dimensional CH equation coupled with a flux BC derived from electrochemical relations to simulate the intercalation of Li^+ into $FePO_4$ material. The authors then depth-average their CHR initial boundary value problem (IBVP) over the spatial coordinate

corresponding to the direction of rapid diffusion in $LiFePO_4$, to obtain a one-dimensional IBVP. This one-dimensional IBVP is second-order (compared with the original fourth-order two-dimensional CHR IBVP) and is more amenable to examination using analytic and numerical methods. Burch [2] extends this work (and examines the assumptions necessary to depth-average in more detail) and shows that the introduction of Cahn-Hilliard dynamics coupled with a reaction term appears to explain recent results from the experimental literature, including the observed reduction of the miscibility gap between phases with decreasing crystal size [10, 11]. Burch [2] also presents a numerical method for solving the full one and two-dimensional CHR IBVPs, though we note that there are errors in their numerical approach that we examine below.

A variety of different numerical techniques have been employed to solve the CH equation, including finite element [12, 13], finite difference [14, 15], spectral [16], boundary integral [17], level set [18], discontinuous Galerkin [19] and multigrid methods [20]. To our knowledge, only Cueto-Felgueroso and Peraire [21] and Burch [2] have investigated the use of the finite volume method (FVM) [22] for solving the CH and CHR equations, respectively. Equations discretised with the FVM obey conservation laws, which make the method well suited to solving CH equations, though including BCs for higher-order PDEs can be difficult.

Cueto-Felgueroso and Peraire [21] use a "dual continuous/discontinuous" moving-least-squares (MLS) approach to estimate information at quadrature points on an unstructured (cell-centred) grid. The MLS method comprises a set of basis functions that are fit (in the least-squares sense) locally using a series of neighbouring node points. These locally fit approximations are then smoothed together to form a globally continuous solution (that includes BCs by using "ghost-nodes" [23]). This global solution is evaluated to obtain flux terms on control volume (CV) faces, though only for elliptic/parabolic terms. For hyperbolic terms, the global solution provides derivative estimates directly at node points, and these are then used to build a separate approximation (based on Taylor series) over a single control volume. It is in this sense that the solution technique is continuous/discontinuous, as the Taylor series used over a single CV will not necessarily be continuous with its neighbours. This is a sophisticated approach that was originally used to tackle shallow water problems [24].

Burch [2] adopts a simpler approach and constructs difference equations to represent the derivatives required at CV mid-points on a (cell-centred) grid. This approach is simple to perform on a linear, orthogonal grid in one dimension, but becomes significantly more difficult in higher dimensions on unstructured grids. Importantly though, the difference equations used near boundaries include the variational boundary condition (VBC). This ensures that all the BCs are included in the discretised fourth-order problem, though again we note that outcomes from this scheme are not presented.

In this work, we present a method for the numerical solution of the CH and CHR problems that incorporates the VBC into a local least-squares approximation based on Taylor series to use with the (vertex-centred) FVM. This simple approach is flexible enough to compute derivatives on an unstructured grid without the need to build a global solution that includes BCs. We then solve the CH and CHR equations in both one and two dimensions applied to Li^+ intercalation in $LiFePO_4$ as a relevant case study, and investigate the role that including the VBC in the least-squares problem has on the numerical solution.

2 Model equations

2.1 Cahn-Hilliard equation

We adopt the notation of Burch [2] and assume that the free energy in our system is given by the CH functional [3] $G[c(\mathbf{x},t)]$ (J), where $c(\mathbf{x},t) \in [0,1]$ is the nondimensional concentration of Li^+ in a $LiFePO_4$ crystal, with spatial coordinates $\mathbf{x}$ (m), through time t (s). This concentration has been nondimensionalised by ρ (m^{-3}), the constant site density (number of molecules per unit volume in the lattice) of Li^+ in a single crystal. For a more detailed derivation of the following model equations, please see [2, 3]. On an N_D-dimensioned domain Ω with boundary Γ the free energy in a CH model can be written as

$$G(t) \equiv G[c(\mathbf{x},t)] = \int_\Omega \left[g_{\text{hom}} + \frac{1}{2}(\nabla c)^{\mathrm{T}} \mathbf{K} (\nabla c) \right] \rho \, \mathrm{d}\Omega, \tag{1}$$

where g_{hom} (J) is the free energy per molecule of a homogeneous system at a uniform concentration and $\mathbf{K}$ (J m^2) is a gradient penalty tensor which we have assumed is symmetric positive definite, isotropic and constant. We assume g_{hom} obeys a regular solution model [2, 3, 25] and is written as

$$g_{\text{hom}}(c) = \Omega_{\text{m}} c(1-c) + k_{\text{B}} T \left(c \log(c) + (1-c)\log(1-c) \right), \tag{2}$$

where k_{B} (J K^{-1}) is Boltzmann's constant, T (K) is the temperature and Ω_{m} (J) is the enthalpy of mixing per site. The chemical potential μ (J) per molecule of Li^+ in the crystal is given by the variational derivative of (1) [26]

$$\mu = \frac{\partial g_{\text{hom}}}{\partial c} - \nabla \cdot (\mathbf{K} \nabla c), \tag{3}$$

which, given the form of (2), can be written as

$$\mu = \Omega_{\text{m}}(1-2c) + k_{\text{B}} T \log\left(\frac{c}{1-c} \right) - \nabla \cdot (\mathbf{K} \nabla c). \tag{4}$$

The mass flux $\mathbf{j}$ (m^{-2} s^{-1}) per molecule is proportional to a gradient in chemical potential, namely

$$\mathbf{j} = \rho c \mathbf{M} \nabla \mu, \tag{5}$$

where $\mathbf{M}$ (m^2 J^{-1} s^{-1}) is a mobility tensor. Finally, as mass is conserved in this system, the Cahn-Hilliard equation is given by

$$\frac{\partial c}{\partial t} + \frac{1}{\rho} \nabla \cdot \mathbf{j} = 0, \qquad \mathbf{x} \in \Omega. \tag{6}$$

The CH equation (6) can also be rewritten directly in terms of the concentration $c(\mathbf{x},t)$ and using the Einstein relation to write the mobility as $\mathbf{M} = \mathbf{D}/(k_{\text{B}}T)$, where $\mathbf{D}$ (m^2 s^{-1}) is a diffusion tensor, we have

$$\frac{\partial c}{\partial t} + \nabla \cdot \left(\mathbf{D} \left\{ \left[2\tilde{\Omega}_{\text{m}} c - \frac{1}{1-c} \right] \nabla c + c \nabla [\nabla \cdot (\tilde{\mathbf{K}} \nabla c)] \right\} \right) = 0, \tag{7}$$

where the tilde notation represents a scaling by $k_{\text{B}}T$ (i.e., $\tilde{\Omega}_{\text{m}} = \Omega_{\text{m}}/k_{\text{B}}T$, $\tilde{\mathbf{K}} = \mathbf{K}/k_{\text{B}}T$). The solution to (7) is the concentration distribution of Li^+ that

minimises the free-energy functional (1) at any time. The initial concentration of Li^+ in a crystal is given by some function $f(\mathbf{x})$, such that

$$c(\mathbf{x}, 0) = f(\mathbf{x}), \qquad \text{at } t = 0, \tag{8}$$

and the boundary conditions for the problem are [2]

$$\hat{\mathbf{n}} \cdot (\tilde{\mathbf{K}} \nabla c) = 0, \qquad \text{on } \Gamma \tag{9}$$

$$\hat{\mathbf{n}} \cdot (\nabla(\nabla \cdot (\tilde{\mathbf{K}} \nabla c))) = 0, \qquad \text{on } \Gamma \tag{10}$$

where $\hat{\mathbf{n}}$ is an outward facing unit normal on Γ. We note that (7) together with (8) to (10) represent the Cahn-Hilliard initial boundary value problem, which we shall refer to as the CH IBVP throughout this work.

Equation (9) is often known as the variational boundary condition (VBC), and Burch [2] emphasises that this condition must be satisfied for (3) to be valid. Together, the boundary conditions (9) and (10) imply the no-flux condition $\hat{\mathbf{n}} \cdot \mathbf{j} = 0$ on Γ. This, along with the form of (6) enforces conservation of mass on the system, or

$$\frac{\mathrm{d}M}{\mathrm{d}t} = \frac{1}{\Omega} \frac{\mathrm{d}}{\mathrm{d}t} \left(\int_\Omega c(\mathbf{x}, t) \, \mathrm{d}\Omega \right) = 0, \tag{11}$$

where $M(t)$ is the dimensionless mass fraction (or average concentration) of Li^+ in the system at time t. In addition, the total free energy of the system must decrease through time [27] and hence

$$\frac{\mathrm{d}\tilde{G}}{\mathrm{d}t} = \int_\Omega \tilde{\mu} \frac{\partial c}{\partial t} \rho \, \mathrm{d}\Omega < 0. \tag{12}$$

Equations (11) and (12) represent physical conditions that should be satisfied by any solution of the CH equation; a numerical scheme that satisfies (12) is known as gradient stable [15, 27].

2.2 Cahn-Hilliard-reaction equation

Singh et al. [1] modify the traditional CH system described in Section 2.1 by including a reaction condition on the boundary. The VBC (9), however, must still be applied, giving the boundary conditions for the CHR problem to be

$$\begin{aligned} \hat{\mathbf{n}} \cdot (\tilde{\mathbf{K}} \nabla c) &= 0, && \text{on } \Gamma \\ \hat{\mathbf{n}} \cdot \mathbf{j} &= \rho_\mathrm{s} R(c), && \text{on } \Gamma \end{aligned} \tag{13}$$

where ρ_s (m^{-2}) is the surface site density and $R(c)$ (s^{-1}) is a reaction term that determines the number of molecules of Li^+ which intercalate into the crystal. Equations (7) to (9) and (13) represent the IBVP that we shall refer to as the CHR IBVP throughout this work. The reaction term $R(c)$ is a Butler-Volmer expression [28], defined in terms of the chemical potential (3) at the surface of the crystal. Burch and Bazant [10] write the general form of this equation as

$$R(c) = R_\mathrm{ins} \left[1 - \exp\left(\tilde{\mu}(c) - \tilde{\mu}_\mathrm{e} \right) \right], \tag{14}$$

where $\tilde{\mu}_\mathrm{e}$ is the nondimensional chemical potential of the electrolyte surrounding the crystal and R_ins (s^{-1}) is the rate for the insertion reaction. For our numerical tests, we keep the general form of (14), but we note that if the CHR IBVP were used in a traditional battery model, (14) could be rewritten to explicitly contain an external overpotential and electrolyte concentration.

3 Finite volume discretisation

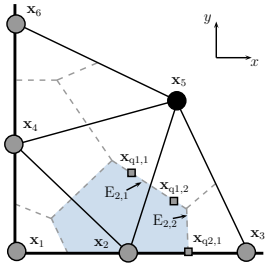


Figure 1: Example unstructured grid near a boundary in 2D. CVs (vertex-centered) are bordered with a dashed line and have node points defined at their centre. Node x_5 is the only node not lying directly on a boundary. $\mathbf{x}_{\mathrm{q}1,1}$ and $\mathbf{x}_{\mathrm{q}1,2}$ are quadrature points on the face $\mathrm{E}_{2,1}$ and $\mathbf{x}_{\mathrm{q}2,1}$ is a quadrature point on face $\mathrm{E}_{2,2}$, that also lies on a boundary.

The finite volume method [22] (FVM) discretises the integral form of a conservation law over a set $\mathcal{V}$ of non-overlapping control volumes (CVs, see Fig. 1); the resulting discrete equations satisfy the original conservation laws exactly. On the ith CV in $\mathcal{V}$, V_i, we define the set of j faces (edges) that border the CV as $\mathcal{E}$, with individual elements in $\mathcal{E}$ written as E_{ij}. The centroid node point in V_i is denoted as as $\mathbf{x}_i$ and the set of quadrature points on the jth element of $\mathcal{E}$ as $\mathcal{X}_{\mathrm{q}}$, with individual quadrature points written as $\mathbf{x}_{\mathrm{q}jk}$. As an example, if we define $i = 2$ for the shaded control volume in Fig. 1, with borders that include the edges $E_{2,1}$ and $E_{2,2}$, $\mathbf{x}_2$ is the centroid node point. Fig. 1 also shows two quadrature point $\mathbf{x}_{\mathrm{q}1,1}$ and $\mathbf{x}_{\mathrm{q}1,2}$, defined on the edge $E_{2,1}$.

We note that (7) is defined in conservative form, so we can integrate over the ith (vertex-centered) CV, V_i, and apply the divergence theorem to obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_{V_i}} c\,\mathrm{d}\Omega_{V_i} = -\oint_{\Gamma_{V_i}} (\mathbf{j}\cdot\hat{\mathbf{n}})\,\mathrm{d}\Gamma_{V_i},$$

where $\hat{\mathbf{n}}$ is an outward facing normal on Γ_{V_i}. We then define an average concentration over the ith CV as

$$\bar{c_i} = \frac{1}{\Omega_{V_i}}\int_{\Omega_{V_i}} c\,\mathrm{d}\Omega_{V_i},$$

and recover the exact form

$$\frac{\mathrm{d}\bar{c}_i}{\mathrm{d}t} = \frac{-1}{\Omega_{V_i}}\oint_{\Gamma_{V_i}} (\mathbf{j}\cdot\hat{\mathbf{n}})\,\mathrm{d}\Gamma_{V_i}. \tag{15}$$

We can then discretise the line integral in (15) to obtain

$$\frac{\mathrm{d}\bar{c}_i}{\mathrm{d}t} \approx \frac{-1}{\Omega_{V_i}}\sum_{j\in\mathcal{E}}\sum_{k\in\mathcal{X}_{\mathrm{q}}} (\mathbf{j}\cdot\hat{\mathbf{n}})_k\, w_k, \tag{16}$$

where w_k are quadrature weightings at the kth quadrature point in $\mathcal{X}_{\mathrm{q}}$ on the jth edge in $\mathcal{E}$. We choose to approximate $\bar{c}_i$ with the value at the ith CV

centroid, which we designate as c_i. This then leaves us to approximate $\mathbf{j} \cdot \hat{\mathbf{n}}$ at each of the quadrature points on any given CV face. We use the least-squares technique described in the next section to accomplish this.

4 Least Squares

Since $\mathbf{j}$ is a function of c and its derivatives (see (4) & (5)), then in order to implement the FVM as described in Section 3 we require the value of c and its derivatives at quadrature points on CV faces. For a general unstructured grid, this can be quite challenging. Pasdunkorale and Turner [29] use a least-squares approach to estimate first derivatives in order to solve a highly anisotropic diffusion problem on an irregular grid. We adopt this basic approach here and a brief overview is given below (see also [30, 31]).

For any node point $\mathbf{x}_i$, we can write a truncated pth order Taylor expansion of the function f, centred about any quadrature point $\mathbf{x}_{\mathrm{q}jk}$ as

$$f(\mathbf{x}_{\mathrm{q}jk} + \mathbf{h}_k) \approx \sum_{d=0}^{p} \frac{1}{d!} (\mathbf{h}_k \cdot \nabla)^d f(\mathbf{x}_{\mathrm{q}jk}), \tag{17}$$

where $\mathbf{h}_k = \mathbf{x}_i - \mathbf{x}_{\mathrm{q}jk}$. The number of unknown components in (17) we require at $\mathbf{x}_{\mathrm{q}jk}$ (f and its derivatives) is given by

$$\begin{aligned} m(p) &= \sum_{d=0}^{p} \binom{N_D + d - 1}{d}, \\ &= \frac{(N_D + p)!}{N_D! p!}. \end{aligned} \tag{18}$$

For any given quadrature point, we can apply (17) to a spread of r nearby node points, contained in the set $\mathcal{X}_{\mathrm{s}}$, whose ith element we refer to as $\mathbf{x}_{\mathrm{s}i}$. We then form an over-determined linear system ($r > m(p)$) for $f(\mathbf{x}_{\mathrm{q}jk})$ and its derivatives, namely

$$\mathbf{WHg} = \mathbf{Wf}. \tag{19}$$

where $\mathbf{H} \in \mathbb{R}^{r \times m(p)}, \mathbf{g} \in \mathbb{R}^{m(p) \times 1}$ and $\mathbf{f} \in \mathbb{R}^{r \times 1}$. The matrix $\mathbf{W} \in \mathbb{R}^{r \times r} = \mathrm{diag}(w_1, w_2, \dots w_r)$, weights nodes closest to the quadrature point, where typically $w_r = ||\mathbf{h}_k||^{-\gamma}$ with $\gamma \in \{1, 2\}$. (Belward et al. [30] note that this weighting does not change the solution, but it improves the numerical conditioning of the system). The least-squares solution of (19),

$$\mathbf{S}\hat{\mathbf{g}} = \min_{\mathbf{g} \in \mathbb{R}^{m(p)}} ||\tilde{\mathbf{W}}\tilde{\mathbf{H}}\mathbf{g} - \tilde{\mathbf{W}}\mathbf{f}||_2, \tag{20}$$

then gives us our approximations at $\mathbf{x}_{\mathrm{q}jk}$. In Eq. (20) we introduced $\tilde{\mathbf{H}} = \mathbf{HS}$, which is equivalent to $\mathbf{H}$ where $\mathbf{h}_k$ is scaled by the maximum distance between $\mathbf{x}_{\mathrm{q}jk}$ and any of the node points used to build $\mathbf{H}$, or $\tilde{\mathbf{h}}_{\mathrm{k}} = \mathbf{h}_k / ||\mathbf{h}_{\max}||$, where $\mathbf{h}_{\max} = \max ||\mathbf{h}_k||$. We also use the scaled $\tilde{\mathbf{h}}_{\mathrm{k}}$ to build $\tilde{\mathbf{W}}$. The matrix $\mathbf{S} \in \mathbb{R}^{m(p) \times m(p)} = \mathrm{diag}(1, ||\mathbf{h}_{\max}||^{-1}, \dots ||\mathbf{h}_{\max}||^{-2}, \dots)$ is applied after calculating the least-squares solution to improve the conditioning of the system, as opposed to solving $\tilde{\mathbf{W}}\tilde{\mathbf{H}}\mathbf{S}^{-1}\mathbf{g} = \tilde{\mathbf{W}}\mathbf{f}$ directly. For the CH equation, we require estimates of up to third derivatives (i.e., $p \geq 3$) and we would like these derivatives to be at

least second-order accurate (i.e., $p \geq 4$). In general, as we increase p to improve the accuracy of our approximations, we are also solving for higher derivative terms that go unused. As such, if we use a given p value in (17), but strictly only need a value of p_o to recover the derivatives we use in the discretisation, we follow Belward et al. [30] and remove the unused $m(p) - m(p_o)$ higher-order derivative terms from our system. Rewriting (19) as

$$\mathbf{W}\mathbf{H}^*\mathbf{g}^* = \mathbf{W}\mathbf{f} - \mathbf{W}\mathbf{M}\mathbf{z}, \tag{21}$$

gives $\mathbf{H}^* \in \mathbb{R}^{r \times m(p_o)}, \mathbf{M} \in \mathbb{R}^{r \times (m(p) - m(p_o))}$, $\mathbf{z} \in \mathbb{R}^{(m(p)-m(p_o)) \times 1}$ and hence $\mathbf{g}^* \in \mathbb{R}^{m(p_o) \times 1}$ only contains the information we explicitly use. We then perform an orthogonal reduction (QR factorisation [32]) on $\mathbf{WM}$, namely $\mathbf{Q}^{\mathrm{T}}\mathbf{WM} = \mathbf{T}$, where $\mathbf{Q}^{\mathrm{T}} \in \mathbb{R}^{r \times r}$ and $\mathbf{T} \in \mathbb{R}^{r \times (m(p)-m(p_o))}$ is upper trapezoidal [23]. This allows us to write (21) in terms of the unused higher-order derivatives by applying $\mathbf{Q}^{\mathrm{T}}$ on the left and hence,

$$\mathbf{Q}^{\mathrm{T}}\mathbf{W}\mathbf{H}^*\mathbf{g}^* = \mathbf{Q}^{\mathrm{T}}\mathbf{W}\mathbf{f} - \mathbf{T}\mathbf{z}. \tag{22}$$

The upper trapezoidal form of $\mathbf{T}$ allows us to discard the first $(m(p) - m(p_o))$ equations in (22) to obtain

$$\mathbf{P}\mathbf{g}^* = \mathbf{f}^*. \tag{23}$$

where now $\mathbf{P} \in \mathbb{R}^{(r-(m(p)-m(p_o))) \times m(p_o)}$ and $\mathbf{f}^* \in \mathbb{R}^{(r-(m(p)-m(p_o))) \times 1}$. Again, using a scaled version of $\mathbf{P}$ and hence $\mathbf{S}$, the least-squares solution of (23)

$$\mathbf{S}\hat{\mathbf{g}}^* = \min_{\mathbf{g}^* \in \mathbb{R}^{m(p_o)}} ||\tilde{\mathbf{P}}\mathbf{g}^* - \mathbf{f}^*||_2, \tag{24}$$

gives us our high-order derivative approximations ((23) will always be over-determined), where now $\mathbf{S} \in \mathbb{R}^{m(p_o) \times m(p_o)}$. Belward et al. [30] showed that the solution to this reduced system is identical to the solution of (19). The benefit of this reduction is that the size of the least-squares problem we must solve has been reduced and hence the solution can be computed more efficiently. This is especially significant as $(m(p) - m(p_o))$ grows, which is the case when we are computing higher order derivatives to high accuracy (and also as N_D increases). We compute the solution to (23) by performing a QR factorisation and solving the resulting (determined) upper triangular system. Both $\tilde{\mathbf{P}}$ and its QR factorisation at every quadrature point can be precomputed (as they only depend on the mesh), and hence $\mathbf{g}^*$ can be obtained relatively cheaply.

To illustrate the algebra performed above, let us consider a specific example with $N_D = 2$, $p = 4$. For simplicity, we use the unscaled $\mathbf{h}_k = (\Delta x_k, \Delta y_k)^{\mathrm{T}}$ and $\mathbf{W} = \mathbf{I}$ in this example. Hence (19) becomes

$$\begin{pmatrix} 1 & \Delta x_{s1} & \Delta y_{s1} & \dots & \frac{\Delta x_{s1}\Delta y_{s1}^3}{6} & \frac{\Delta y_{s1}^4}{24} \\ 1 & \Delta x_{s2} & \Delta y_{s2} & \dots & \frac{\Delta x_{s2}\Delta y_{s2}^3}{6} & \frac{\Delta y_{s2}^4}{24} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Delta x_{sr} & \Delta y_{sr} & \dots & \frac{\Delta x_{sr}\Delta y_{sr}^3}{6} & \frac{\Delta y_{sr}^4}{24} \end{pmatrix} \begin{pmatrix} f(\mathbf{x}_{\mathrm{q}jk}) \\ \frac{\partial f}{\partial x}(\mathbf{x}_{\mathrm{q}jk}) \\ \vdots \\ \frac{\partial^4 f}{\partial y^4}(\mathbf{x}_{\mathrm{q}jk}) \end{pmatrix} = \begin{pmatrix} f(\mathbf{x}_{s1}) \\ f(\mathbf{x}_{s2}) \\ \vdots \\ f(\mathbf{x}_{sr}) \end{pmatrix}, \tag{25}$$

where $\mathbf{H} \in \mathbb{R}^{r \times 15}, \mathbf{g} \in \mathbb{R}^{15 \times 1}$ and $\mathbf{f} \in \mathbb{R}^{r \times 1}$. If we wish to eliminate the fourth derivatives from the system, we set $p_o = 3$, giving $m(p_o) = 10$, and the reduction

(21) gives

$$\mathbf{M} = \begin{pmatrix} \frac{\Delta x_{s1}^4}{24} & \frac{\Delta x_{s1}^3 \Delta y_{s1}}{6} & \frac{\Delta x_{s1}^2 \Delta y_{s1}^2}{4} & \frac{\Delta x_{s1} \Delta y_{s1}^3}{6} & \frac{\Delta y_{s1}^4}{24} \\ \frac{\Delta x_{s2}^4}{24} & \frac{\Delta x_{s2}^3 \Delta y_{s2}}{6} & \frac{\Delta x_{s2}^2 \Delta y_{s2}^2}{4} & \frac{\Delta x_{s2} \Delta y_{s2}^3}{6} & \frac{\Delta y_{s2}^4}{24} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\Delta x_{sr}^4}{24} & \frac{\Delta x_{sr}^3 \Delta y_{sr}}{6} & \frac{\Delta x_{sr}^2 \Delta y_{sr}^2}{4} & \frac{\Delta x_{sr} \Delta y_{sr}^3}{6} & \frac{\Delta y_{sr}^4}{24} \end{pmatrix}; \qquad \mathbf{z} = \begin{pmatrix} \frac{\partial^4 f}{\partial x^4}(\mathbf{x}_{\mathrm{q}jk}) \\ \frac{\partial^4 f}{\partial x^3 y}(\mathbf{x}_{\mathrm{q}jk}) \\ \vdots \\ \frac{\partial^4 f}{\partial y^4}(\mathbf{x}_{\mathrm{q}jk}) \end{pmatrix}, \tag{26}$$

where $\mathbf{M} \in \mathbb{R}^{r \times 5}$ and $\mathbf{z} \in \mathbb{R}^{5 \times 1}$. After performing the orthogonal reduction and removing the first $m(p) - m(p_o) = 5$ equations from (22), our smaller least-square problem (23) now has dimensions of $\mathbf{P} \in \mathbb{R}^{(r-5) \times 10}, \mathbf{g}^* \in \mathbb{R}^{10 \times 1}$ and $\mathbf{f}^* \in \mathbb{R}^{(r-5) \times 1}$.

4.1 Variational Boundary Condition

For quadrature points away from the boundaries of the domain, the method described in Section 4 works well. However, near boundaries, we face challenges in implementing the BCs. Namely, in a finite volume framework we impose conditions directly on the boundary fluxes, and so it can be difficult to include BCs that are not in this form. We face additional difficulties given the fourth-order CH equation described in Section 2. For example, in one dimension using planar coordinates, the CH flux (5) takes the form

$$\mathbf{j}_x = \mathbf{D}_{xx} \left(\left[2\tilde{a}c - \frac{1}{1-c} \right] \frac{\partial c}{\partial x} + c \tilde{\mathbf{K}}_{xx} \frac{\partial^3 c}{\partial x^3} \right), \tag{27}$$

and on Γ, (9) and (10) reduce to

$$\tilde{\mathbf{K}}_{xx} \frac{\partial c}{\partial x} = 0; \qquad \tilde{\mathbf{K}}_{xx} \frac{\partial^3 c}{\partial x^3} = 0. \tag{28}$$

which, of course implies $\mathbf{j}_x = 0$ on Γ, but imposing $\mathbf{j}_x = 0$ on (27) does not necessarily imply (28).

Burch [2] tackled this problem by substituting the VBC into the standard analytic difference equations used to evaluate information at CV faces near boundaries. Any face whose spread includes a node point that lies on the boundary must use these new difference equations. Quantities evaluated at these faces therefore explicitly include the VBC and Burch [2] is then free to set a flux condition at the boundary of the domain in the standard manner.

The difference equations given by Burch [2] can be derived analytically in traditional ways; perhaps the easiest in this circumstance is to construct an interpolating polynomials using the Newton divided difference form [33]. For a face near a boundary in one-dimension, an extra row is added to the linear system expressing that the derivative of the polynomial is zero at the boundary. The resulting polynomial can then be evaluated/differentiated at CV faces. We note this must be performed for all the quantities $(c, \nabla c, \dots)$ required at every face that uses boundary node information. This is relatively simple in one-dimension with a linear mesh, however in higher dimensions and with unstructed grids this process become difficult. We also note that the difference equations constructed by Burch [2] for mixed higher derivatives in two-dimensions are incorrect (successive univariate approximation [33] cannot be applied near corners when including the VBC).

Cueto-Felgueroso and Peraire [21] use a cell-centred FVM to solve the CH equation. They overcome the BC problem by creating extra node points on the boundary of the domain (that would already exist in a vertex-centred approach) whose function values are extrapolated from the BCs (similar to "ghost nodes" [23]). These extra nodes are then included in the spread used to construct the global MLS approximation in order to give "stronger enforcement of the BCs". As mentioned in Section 1, Cueto-Felgueroso and Peraire [21] evaluate hyperbolic terms by using the global MLS approximation to calculate derivative values at node points. Taylor series are then constructed (locally) over individual CVs using this information. These series are then evaluated to compute values at quadrature points on each CV (which makes it easier to apply limiting schemes [24, 34]). Given that the global MLS approximation includes the BCs, and the derivatives used to build the Taylor series are evaluated using this global MLS, the hyperbolic terms should include the BCs, though to what degree is unclear.

The approach we take includes the VBC (9) explicitly in our (local) least-squares approach (which, as we have seen in the previous section, is built using Taylor series) at quadrature points near the boundary. In a general sense, BCs are often included in least-squares approximations (see [29, 35] for example). This is normally done when solving advection-diffusion problems in order to to apply BCs that are not necessarily in "flux-form" (like (13)). These conditions are included in the LS system at quadrature points that lie directly on a boundary, which we will denote as $\mathbf{x}^{\mathrm{b}}_{\mathrm{q}jk}$ (e.g., $\mathbf{x}_{\mathrm{q}2,1}$ in Fig. 1). We explicitly include (9) in this manner by adding a single additional equation to (19), namely

$$\hat{\mathbf{n}} \cdot \left(\tilde{\mathbf{K}} \nabla f(\mathbf{x}^{\mathrm{b}}_{\mathrm{q}jk}) \right) = 0. \tag{29}$$

This equation is easily added to the linear system, as it is expressed directly at the quadrature point, and (19) is already over-determined.

Only including (29) in our approximation, however, is not enough to ensure enforcement of the VBC (9) for our 4th-order PDE (7). When constructing the linear system for quadrature points near a boundary, we note that r_{b} of the r nodes used to build the linear system lie on Γ (the grey nodes in Fig. 1). We denote one of these r_{b} points as $\mathbf{x}_{\mathrm{b}}$, and note that we must express the VBC (9) at each of these boundary nodes, namely

$$\hat{\mathbf{n}} \cdot \left(\tilde{\mathbf{K}} \nabla f(\mathbf{x}_{\mathrm{b}}) \right) = 0. \tag{30}$$

We cannot, however, directly include (30) in our LS system like (29), as it does not apply at the quadrature point. We therefore express (30) as a Taylor series centred at the quadrature point of interest, which we denote as $\mathbf{x}_{\mathrm{q}}$ (which could be either $\mathbf{x}_{\mathrm{q}jk}$ or $\mathbf{x}^{\mathrm{b}}_{\mathrm{q}jk}$) and so we can write

$$\hat{\mathbf{n}} \cdot \left(\tilde{\mathbf{K}} \nabla f(\mathbf{x}_{\mathrm{b}}) \right) \approx \left(\hat{\mathbf{n}} \cdot (\tilde{\mathbf{K}} \nabla) \right) \left(\sum_{d=0}^{p-1} \frac{1}{d!} (\mathbf{h}_{\mathrm{b}} \cdot \nabla)^d f(\mathbf{x}_{\mathrm{q}}) \right) = 0, \tag{31}$$

where $\mathbf{h}_{\mathrm{b}} = \mathbf{x}_{\mathrm{b}} - \mathbf{x}_{\mathrm{q}}$. We are now free to include (31) in (19) for each of the r_{b} boundary points (taking care to use the scaled $\tilde{\mathbf{h}}_{\mathrm{b}} = \mathbf{h}_{\mathrm{b}} / ||\mathbf{h}_{\max}||$ where applicable for consistency). We note that this approach would still be applicable if the VBC were non-homogeneous, and if surface effects [2] were included in

(1). For the CH IBVP, we also include (10) in this manner. To illustrate the above approach, let us consider the specific example where we are computing information at the quadrature point $\mathbf{x}_{\mathrm{q}2,1}$ in Fig. 1 and we include a single boundary node in the least-squares system, say $\mathbf{x}_4$. Equation (30) reduces to

$$-\frac{\partial f}{\partial x} = 0 \qquad \text{at } \mathbf{x}_4, \tag{32}$$

which we can express using (31) as

$$-\frac{\partial f}{\partial x}(\mathbf{x}_4) \approx -\left(\frac{\partial f}{\partial x}(\mathbf{x}_{q2,1}) + \Delta x_4 \frac{\partial^2 f}{\partial x^2}(\mathbf{x}_{q2,1}) + \ldots + \frac{\Delta y_4^3}{6}\frac{\partial^4 f}{\partial y^3 \partial x}(\mathbf{x}_{q2,1})\right) = 0. \tag{33}$$

If we consider our example system (25), (19) now becomes

$$\begin{pmatrix} 1 & \Delta x_1 & \Delta y_1 & \ldots & \frac{\Delta x_1 \Delta y_1^3}{6} & \frac{\Delta y_1^4}{24} \\ 1 & \Delta x_2 & \Delta y_2 & \ldots & \frac{\Delta x_2 \Delta y_2^3}{6} & \frac{\Delta y_2^4}{24} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Delta x_r & \Delta y_r & \ldots & \frac{\Delta x_r \Delta y_r^3}{6} & \frac{\Delta y_r^4}{24} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & -1 & 0 & \vdots & -\frac{\Delta y_4^3}{6} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & -1 & \vdots & 0 & 0 \end{pmatrix} \begin{pmatrix} f(\mathbf{x}_{\mathrm{q}2,1}) \\ \frac{\partial f}{\partial x}(\mathbf{x}_{\mathrm{q}2,1}) \\ \vdots \\ \frac{\partial^4 f}{\partial y^4}(\mathbf{x}_{\mathrm{q}2,1}) \end{pmatrix} = \begin{pmatrix} f(\mathbf{x}_{s1}) \\ f(\mathbf{x}_{s2}) \\ \vdots \\ f(\mathbf{x}_{sr}) \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{34}$$

where $\mathbf{H} \in \mathbb{R}^{(r+r_{\mathrm{b}}+1)\times 15}$ and $\mathbf{f} \in \mathbb{R}^{(r+r_{\mathrm{b}}+1)\times 1}$. The first r rows in (34) still express (17) for the spread of r node points that we choose to build the least-square system. The next visible row includes (33) in the linear system, and we note as mentioned above, extra rows (that express (31) like (33)) must be included for each of the r_{b} boundary node points used. The last row in (34) implements (29), as $\mathbf{x}_{q2,1}$ lies on a boundary, specifically

$$-\frac{\partial f}{\partial y} = 0, \qquad \text{at } \mathbf{x}_{q2,1}.$$

This final equation would not be included if we were estimating quantities, for example, at $\mathbf{x}_{\mathrm{q}1,1}$ in Fig. 1, though the r_{b} equations representing (31) at each boundary node point would be. Furthermore, we note that in this work, we solve the equations given in Section 2 on a rectangular domain, and so at a corner node (for example $\mathbf{x}_1$ in Fig. 1) we apply (31) in both the x and y directions separately.

To solve the modified least-squares problems that occur near the boundaries and of which (34) is a specific example, we use two approaches. The first is known as equality-constrained least-squares [36], and involves removing the boundary condition information from $\mathbf{WH}$ and instead writing these conditions as the separate linear system

$$\mathbf{Bg} = \mathbf{d}, \tag{35}$$

where $\mathbf{B} \in \mathbb{R}^{r_{\mathrm{t}}\times m(p)}$, $\mathbf{d} \in \mathbb{R}^{r_{\mathrm{t}}\times 1}$ and r_{t} is the total number of boundary equations we have applied (which depends on the position of the quadrature point, whether

we have included a corner node, and the specific IBVP we are solving). We then seek the least-squares solution to (19), but constrained such that (35) is satisfied exactly, or

$$\mathbf{S}\hat{\mathbf{g}} = \min_{\tilde{\mathbf{B}}\mathbf{g}=\mathbf{d}} ||\tilde{\mathbf{W}}\tilde{\mathbf{H}}\mathbf{g} - \tilde{\mathbf{W}}\mathbf{f}||_2. \tag{36}$$

In solving this constrained system, we do not apply the reduction described by (21), instead we retain the $m(p) - m(p_o)$ higher order terms in the system and apply the nullspace method, which uses a QR factorisation (see [36, 37] for details) to solve the complete higher-order system in (36). The constrained system is only solved at the (small number of) quadrature points near the boundary, and hence the added cost from solving the full system is not great.

The second approach we use notes that the solution to (36) can be approximated [36] by leaving the r_t boundary equations in $\tilde{\mathbf{W}}\tilde{\mathbf{H}}$ and applying a large weighting to these equations. The boundary constraints in (35) are already weighted by $||\tilde{\mathbf{h}}_\mathrm{b}||^{-\gamma}$, so we multiply this weighting by a constant λ. For large λ, the solution of this weighted least-squares problem approximates the solution given by the nullspace method (as long as the size of λ does not significantly affect the conditioning of $\tilde{\mathbf{W}}\tilde{\mathbf{H}}$). The benefit of this weighted approach is that it enables us to easily use the reduction in (21).

The numerical approach introduced in this section for the application of non-standard BCs in the solution of higher-order PDEs shares similarities with that used in immersed (or embedded) boundary methods (see [38] for a recent review), where BCs must be imposed on the boundary of a domain that does not lie on grid lines. Often, an interpolating function that includes the BC is used to approximate values near a boundary [39, 40]; our need to impose boundary conditions in this fashion results from a 4th order PDE, as opposed to a nonbody conformal boundary, though our approach could most likely be used in that context.

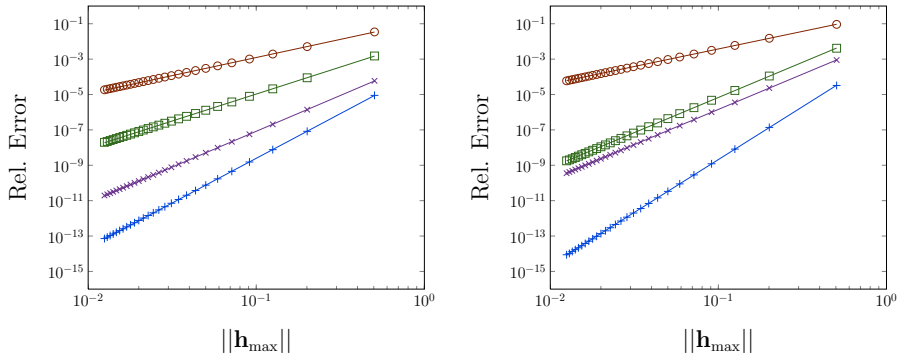
4.2 Test functions

Theoretical error bounds for estimating gradients using the weighted least-square approach are given in Turner et al. [31], and both Belward et al. [30] and Turner et al. [31] provide numerical evidence of second order accuracy in estimating first derivatives for a series of test functions on unstructured grids. In order to confirm the accuracy of our method for computing up to third derivatives with the VBC (9) included, we tested the derivative estimation on two simple test functions in one-dimension, namely

$$f(x) = \sin(x) \qquad x \in [-10, 10], \tag{37}$$

$$f(x) = 0.1 + 0.8\exp(-x^2) \qquad x \in [-10, 10]. \tag{38}$$

Here, (38) is the initial condition that we apply in Section 5 and was used by Singh et al. [1] in their work on the CHR IBVP. We tested the least-squares approach described in Section 4 on (37) and as (38) (numerically) satisfies (9) on the boundaries of the test domain, it provides a convenient way to test the addition of the VBC described in Section 4.1. We impose a linear mesh on the one-dimensional domain, and used the weighted least-squares approach (with boundary weighting $\lambda = 10$ for (38)) to calculate the value of $f(x)$ and up to third derivatives at the CV faces half-way between each node. As we require



(a) $\sin(x)$. Order: $+$: 5.04, $\times$: 4.04, $\square$: 3.04, $\bigcirc$: 2.04.

(b) $0.1 + 0.8\exp(-x^2)$. Order: $+$: 5.97, $\times$: 3.97, $\square$: 3.97, $\bigcirc$: 1.98.

Figure 2: Relative error of approximations versus the maximum distance between a face and the node points used to construct the linear system. Solid lines are least-squares fit to the numerical results and are used to determine the order of each approximation. $+$: c, $\times$: $\partial c/\partial x$, $\square$: $\partial^2 c/\partial x^2$, $\bigcirc$: $\partial^3 c/\partial x^3$

third derivative terms, we set $p_o = 3$ and to ensure these derivatives are second order accurate, we let $p = 4$. We also assign $r = 6$, so that the nearest six node points are used to build the linear system ($m(p) = 4$ in this case). We computed the relative error for f and its derivatives as the grid spacing decreases, and for example, the error in f is measured by

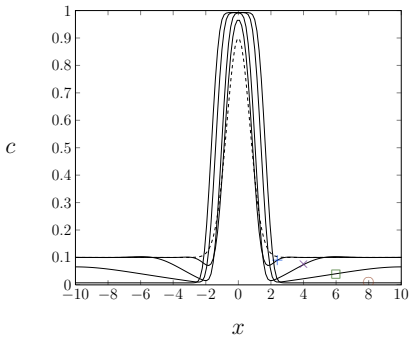
$$\text{Rel. Error} = \frac{||f_{\text{exact}} - f_{\text{approx}}||_\infty}{||f_{\text{exact}}||_\infty}.$$

The results of this test are shown in Fig. 2, with Fig. 2a being a plot of the error for (37) and Fig. 2b being the same, but for (38). We see that in Fig. 2a the relative error in the approximations is decreasing at the expected rate; the 5th order convergence for c aligns with $p = 4$ and $p_o = 3$ (as the elimination of the fourth derivatives should push the remainder term in the Taylor series to 5th order). Each of the higher-order derivative terms lose an order of accuracy, and we see that upon reaching the third derivative our approximation is second order, as desired.

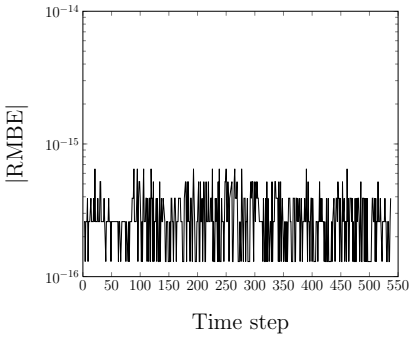
Similar results are shown in Fig. 2b, where the VBC is included at faces near the boundary. We note that both the weighted and equality-constrained least-squares approaches described in Section 4.1 produce identical results for this test. We again see that the third derivatives are approximated to second order. Given these results, we now proceed to investigate the solution of the CH and CHR IBVPs using the least-squares method described above.

5 Results

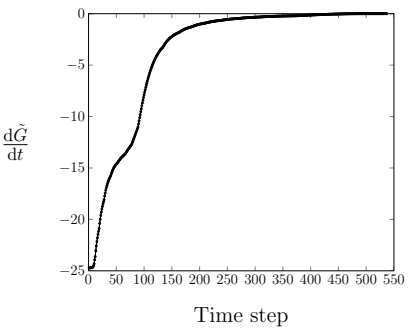
The results presented in this section were computed in C++ and the Intel Compiler (Version 12.0.3.174) was used to compile the code on an Intel Xeon X5650 processor (64-bit with 6 cores running at 2.66 GHz) with the optimisation level set at -O3 and the compiler flag "fp-mode strict".



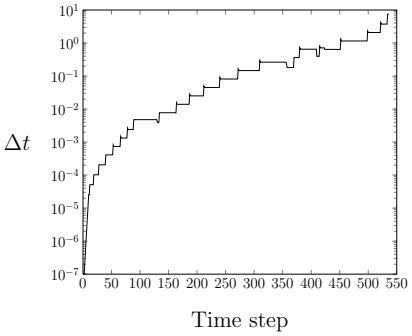
(a) Solution profile, dashed line is the initial condition: $0.1 + 0.8\exp(-x^2)$. t (s): + 0.149, × 3.1, □ 34.5, ○ 179.6.



(b) Relative mass balance error over time.



(c) Change in free energy over time.



(d) Size of the time steps taken

Figure 3: CH IBVP solution computed in one dimension ($x \in [-10, 10]$) with 150 nodes using equality-constrained least-squares. Relative and absolute tolerances were set at 1×10^{-6}. $\tilde{\Omega}_{\mathrm{m}} = 5$, $\tilde{\mathbf{K}} = \mathbf{I}$, $\mathbf{D} = \mathbf{I}$.

We used the SPQR package in SuiteSparse [41] to perform QR factorisations and used the Intel MKL (BLAS, LAPACK and VML) whenever possible. No direct effort was made to parallelise the software produced, however both the SuiteSparse package and the Intel MKL are heavily multi-threaded.

The coupled set of ODEs (in time) generated by the FVM discretisation were solved using the IDA module from Sundials [42]. IDA provides adaptive time stepping with error control using a backward differentiation formula (BDF) (of up to order 5). There have been a number of papers in the literature exploring various time-stepping schemes for the CH equation (see [43, 44, 45, 21] for example), most based on the work of Eyre [27] who showed that unconditionally gradient-stable time-stepping is possible. Here, we are primarily concerned with the spatial discretisation of the CH and CHR equations and as such we do not examine the efficiency of the time-stepping scheme. We set IDA to use a maximum of 2nd order time-stepping to ensure that the numerical solution remains gradient-stable. We also use the sign of $\mathrm{d}\tilde{G}/\mathrm{d}t$ as a recoverable error, which forces IDA to use smaller (order and size) time-steps if gradient-stability is violated.

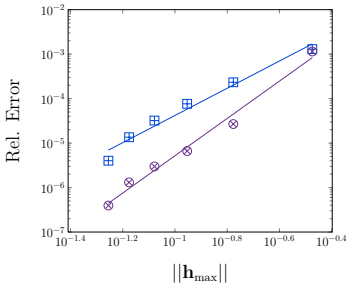
As mentioned above, we impose a linear, orthogonal mesh in both one and

two dimensions. This mesh is imposed to increase the computational efficiency of our solution, as it allows us to estimate information at most of the faces by solving a single least-squares problem with multiple RHS vectors. Such an approach is motivated by our case study, in that our application of CHR dynamics to $LiFePO_4$ requires efficient solutions on regular domains. As we have emphasised throughout this work however, the least-square technique discussed in Section 4 can be used without modification on unstructured grids. With efficiency in mind, we also set quadrature points in our discretisation to lie at the midpoint of CV faces, which results in (16) being second-order. Higher-order approximations to the line integral in (15) rapidly become more costly as the number of quadrature points on a CV face grows, since a least-square problem must be solved at each of these.

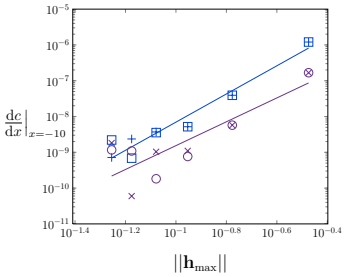
A minimum value of $p = 4$ and $p_o = 3$ was used throughout the simulations presented to ensure that our third-derivative approximations stay second-order (unless otherwise noted). We set $\gamma = -2, \lambda = 10$, and note that this weighting significantly affects the conditioning of $\mathbf{WH}$. This weighting becomes particularly important when solving the CHR IBVP because numerical errors in derivative approximations are exacerbated by the exponential form of the BC (13). In an attempt to minimise this numerical error, we also investigated using a truncated SVD decomposition to construct a pseudo-inverse [32] (PINV in MATLAB) in order to solve the least-squares problem. The results of those tests are not shown here, as the pseudo-inverse did not perform as well as the weighted approach (this is to be expected, as pseudo-inverses are notoriously unstable numerically [32] and no great effort was made to investigate different truncation values). We also set $\tilde{\mathbf{K}} = \mathbf{D} = \mathbf{I}$ and note that for $LiFePO_4$, these tensors would be highly anisotropic. As such, the results presented are not strictly representative of Li^+ intercalation in $LiFePO_4$. We chose to keep the tensors symmetric as again, the exponential form of the BC (13) exacerbates any numerical error and if care is not taken at the boundaries, particularly for the CHR IBVP, asymmetric behaviour results. The results shown below are perfectly symmetric about the centre of the domain, but if any asymmetry were present, this would be obscured by anisotropic tensors.

When solving the CHR IBVP we require the value of the chemical potential $\tilde{\mu}$ at node points on the boundaries. This is given by a suitably scaled version of (4). Furthermore, we note that, given our vertex-centred approach, the only unknown quantity in (4) is the Laplacian of c. We use the least-squares method described above to approximate this value, whereby the boundary conditions are implemented in the same manner as in Section 4.1 for a quadrature point that lies on the boundary. When estimating the Laplacian at one of these boundary node points, a Taylor series approximation (17) is not required to express the node value at this point, as it can be expressed directly in the linear system. We set the weighting at this point to be equal to the weighting at the nearest neighbour multiplied by 100 (as $\mathbf{h}_k = \mathbf{h}_\mathrm{b} = \mathbf{0}$).

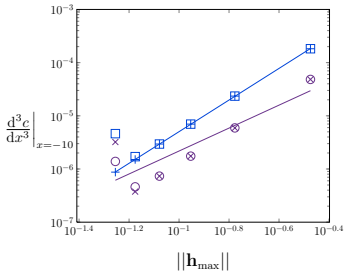
In one-dimension, we use a spread of the nearest 6 points to construct the linear system, while in two-dimensions we use the nearest 30 points. A careful balance is required when choosing the spread of points to include. Enough node points must be used to accurately estimate the information required, however node points that are far from the quadrature point contribute significantly to the singular nature of $\mathbf{H}$, especially as p increases (the smallest singular value of $\mathbf{H}$ appears as a denominator in the error bound presented by [31]).



(a) Relative error in the solution (measured with respect to a reference solution). Order: $+$: 3.06, $\times$: 4.1965.



(b) Maximum absolute value of the the first derivative at $x = -10$ throughout the simulation. Order: $+$: 3.9355, $\times$: 2.9735.



(c) Maximum absolute value of the third derivative at $x = -10$ throughout the simulation. Order: $+$: 3.3319, $\times$: 2.157.

Figure 4: Convergence rates (with respect to a reference solution) for the 1D CH IBVP vs the maximum distance between a face and the node points used to construct the linear system. Relative and absolute tolerances were reduced to 1×10^{-12}. Closed shapes ($\square$, $\bigcirc$) use equality-constrained least-squares near boundaries, open shapes ($+$, $\times$) use weighted least-squares. Solid lines are least-squared fit to the weighted least-square results to determine the order of each approximation. $+$, $\square$: $p = 4, p_o = 3$, $\times$, $\bigcirc$: $p = 5, p_o = 4$.

The Jacobian resulting from this spread is naturally banded, and so we use the globally convergent, banded Newton solver in IDA to solve the nonlinear system. Overall, computing the numerical solution to the CH and CHR IBVPs takes a few seconds in one dimension, and several hours in two dimensions, with moderate tolerances.

5.1 One-dimensional results

5.1.1 CH IBVP

Fig. 3 shows the results from solving the CH IBVP with 150 node points using the equality-constrained, least-squares method. Fig. 3a shows the dimensionless concentration profile through time. We can see that the solution evolves from the symmetric initial condition (8) to form a narrow band of phase separated material that is symmetric about $x = 0$. The solution at $t = 179.6$s represents

the steady state of the problem, in which the material has phase separated as much as possible. Fig. 3b shows the absolute value of the relative mass balance error, which is computed by

$$|\text{RMBE}| = \frac{|M(0) - M(t)|}{|M(0)|}.$$

The mass is computed numerically by using the trapezoid rule to evaluate the integral in (11). We can see that the mass balance error is very close to machine precision throughout the entire simulation; this confirms that our numerical method is conservative, as is expected when using the FVM. We are aware of only one numerical paper for the CH equation that approaches (or shows evidence of) this level of conservation [17]. In contrast, papers on "conservative" finite difference schemes for the CH equation often plot the total mass in the system with very large axis scalings, obscuring the degree to which these schemes conserve [14, 46, 15].

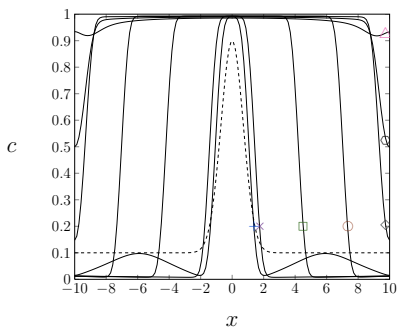
The other physical condition our numerical scheme should satisfy is that of gradient stability, given by (12). Fig. 3c plots the change in free energy over time, $\mathrm{d}\tilde{G}/\mathrm{d}t$, and we see that $\tilde{G}(t)$ is always decreasing throughout the simulation. We note that at no point in Fig. 3c does $\mathrm{d}\tilde{G}/\mathrm{d}t$ become positive, although the rate of decrease in the free energy slows towards the end of the solution (and approaches zero while still remaining negative) as the material phase separates and reaches equilibrium. Normally, papers in the literature plot the value of $\tilde{G}(t)$, as opposed to $\mathrm{d}\tilde{G}/\mathrm{d}t$. We chose to plot the rate of change as it is obvious when $\mathrm{d}\tilde{G}/\mathrm{d}t$ changes sign. Unless the authors explicitly note that their solutions do not stay strictly gradient stable (like Cueto-Felgueroso and Peraire [21]), small increases in free energy are not always evident in a plot of $\tilde{G}(t)$.

Finally, Fig. 3d shows the size of the time-steps taken. The adaptive time-stepping of IDA is clearly evident, as the size of the time-step increases over 7 orders of magnitude throughout the simulation. Again, as mentioned previously in Section 5, the objective of this paper is not to investigate different time-stepping schemes, and we acknowledge that there are schemes that achieve solutions using a smaller number of time-steps than the 550 shown in Fig. 3d (for example [21]).

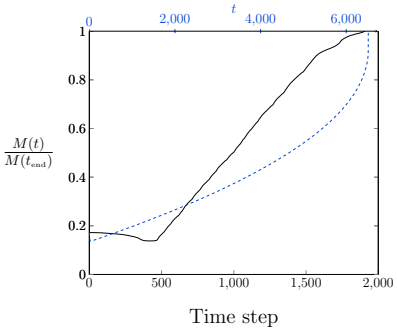
Now that we have shown that the numerical solution satisfies the relevant physical conditions, we turn to Fig. 4 which shows the convergence rates of the our scheme. Specifically, Fig. 4a plots the relative error of the solution as the grid spacing decreases. We measure the relative error by

$$\text{Rel. Error} = \frac{||c_{\text{rep}} - c_{\text{approx}}||_\infty}{||c_{\text{rep}}||_\infty},$$

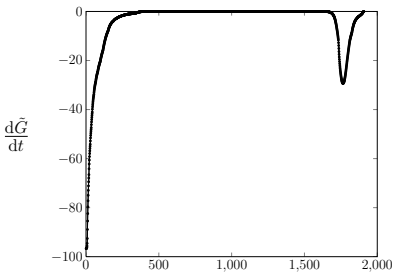
where the representative solution c_{rep} is computed with the equality-constrained least-squares method on a linear mesh with 1000 node points, relative and absolute tolerances set to 1×10^{-12}, $p = 5$ and $p_o = 4$. Node points at the same spatial coordinate in the representative solution are then used in the error calculation. We also tested linearly interpolating the representative solution onto the courser mesh and found that the greatest error always occurs at the boundaries of the domain, and so both methods produce the same error. In Section 4.2, we verified that setting $p = 4, p_o = 3$ produces second-order accurate third derivatives, and so we expect the numerical solution to the CH equation should be at



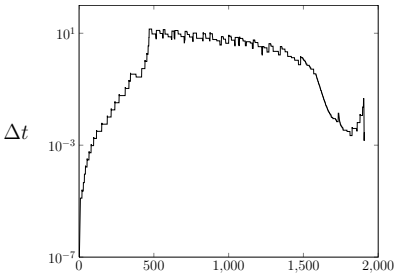
(a) Solution profile, dashed line is the initial condition: $0.1+0.8\exp(-x^2)$. t (s): + 4.72, × 266, □ 3920, ○ 6010, ◇ 6515.194, ⬡ 6515.197, △ 6515.8.



(b) Proportion of mass in the system over time. The solid line corresponds to the bottom x axis (time step), while the dashed line corresponds to the top x axis (t).



(c) Change in free energy over time.



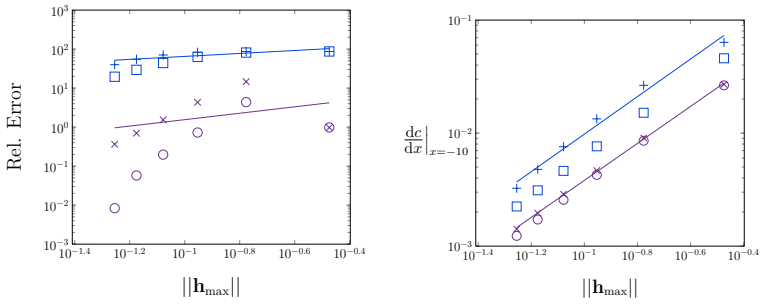
(d) Size of the time steps taken.

Figure 5: CHR IBVP solution computed in one dimension ($x \in [-10, 10]$) with 150 nodes using equality-constrained least-squares. Relative and absolute tolerances were set at 1×10^{-6}. $\tilde{\Omega}_{\mathrm{m}} = 5$, $\tilde{\mathbf{K}} = \mathbf{I}$, $\mathbf{D} = \mathbf{I}$, $\rho_{\mathrm{s}} R_{\mathrm{ins}}/\rho = 1$, $\tilde{\mu}_e = 0.5$.

least-second order accurate. We see, however in Fig. 4a that the solution is converging at a third order rate. We also computed the error for the higher-order case, with $p = 5, p_o = 4$ and we note this solution is approximately fourth-order. Fig. 4a also shows that the solutions produced by the equality-constrained and weighted least-square methods are identical and converging at the same rates.

Importantly, Fig. 4b and 4c show the degree to which the numerical solution satisfies the BCs. Fig. 4b plots the value of the gradient at the left boundary, which was measured using a simple sixth order difference equation [47], while Fig. 4c plots the third derivative at the same boundary, measured using a second order difference equation [47]. We only plot the values at the left boundary as they are identical to the values on the right boundary, given the symmetric initial condition. In a general sense, how well the BCs are satisfied depends on the grid spacing, as we are including the BCs in the least-square problem using a Taylor series centred away from the boundary.

In Fig. 4b we see that, for the $p = 4$ case, we satisfy $\partial c/\partial x = 0$ to approximately fourth order for the weighted least-squares (the fit lines for equality-constrained are not shown, they give very similar rates of convergence). The equality-constrained and weighted least-squares approximations also give



(a) Relative error in the solution (measured with respect to a reference solution). Order: +: 0.38, ×: 0.82, □: 0.80, ○: 2.65 (fit lines not shown for □, ○).

(b) Maximum absolute value of the first derivative at $x = -10$ throughout the simulation. Order: +: 1.66, ×: 1.64.

Figure 6: Convergence rates (to a reference solution) for the 1D CHR equation vs the maximum distance between a face and the node points used to construct the linear system. Relative and absolute tolerances were reduced to 1×10^{-12}. Closed shapes (□, ○) use equality-constrained least-squares near boundaries, open shapes (+, ×) use weighted least-squares. Solid lines are least-squared fit to the weighted least-square results to determine the order of each approximation. +, □: $p = 4, p_o = 3$, ×, ○: $p = 5, p_o = 4$.

slightly different values as the grid spacing becomes small (near 600 nodes). This is also the case for $p = 5$, where we see that the weighted least-squares method converges at a rate of around 3. The equality-constrained least-squares however, performs better in this circumstance.

The behaviour shown in Fig. 4b, where the satisfaction of the BC begins to vary as the grid spacing decreases is due to the singular nature of the problem as both p increases and the grid spacing decreases. An example of this singular behaviour can be observed by computing the condition number of the unscaled $\mathbf{WH}$ (with boundary conditions included as per the weighted least-squares). With 150 node points and $p = 4$ the condition number is 6.67×10^4, whereas for $p = 5$, it increases to 1.17×10^6. This shows that the system becomes increasingly ill-conditioned at higher orders. This is also true as the number of node points increases, as we would be expect. We should note that the condition number of the scaled matrix $\tilde{\mathbf{W}}\tilde{\mathbf{H}}$ doesn't change with grid spacing, however we are still solving an equivalent system and simply applying $\mathbf{S}$ after the solve in an attempt to overcome some of these conditioning issues.

Fig. 4c shows that $\partial^3 c/\partial x^3 = 0$ is satisfied to slightly over third order for $p = 4$ and second order for $p = 5$ (though the second-order difference equation used to measure the third derivative may be masking a slightly higher-order convergence rate). Again, as the grid spacing becomes small (near 900 nodes), we see the data points become slightly erratic, though it appears that the weighted least-squares performs better than the equality-constrained in this case.

In summary, it appears that for the CH IBVP, our numerical method performs well for low to moderate grid spacing (up to approximately 600 nodes), in that it satisfies all of the relevant physical conditions (such as mass conservation and gradient stability), along with implementing the BCs in a high-order

manner. We now investigate the solution to the CHR IBVP.

5.1.2 CHR IBVP

Fig. 5 shows the results from solving the CHR IBVP with 150 node points using the equality-constrained least-squares method. We can see in Fig. 5a that initially (up to $t = 266$s), the solution profile phase-separates in a similar manner to the no-flux case (though the removal of the condition on the third derivative means a small hump forms at $x = \pm 6$ before fully phase-separating). A travelling wave then forms and moves through the domain, before impacting the boundaries, and filling to a uniform concentration. Interestingly, Fig. 5b shows that the total mass of the system decreases initially (the 500th time-step corresponds to $t \approx 266$s), before increasing for the rest of the simulation. The behaviour of the CHR IBVP is extremely complex; this deintercalation for example, is driven by the form of (13). The chemical potential at the boundaries of the solid μ, must be greater than the external chemical potential μ_{e} to drive flux into the solid.

Burch [2] notes that unlike the CH equation there is no analytic proof that the solution to the CHR IBVP should remain gradient-stable. However, our numerical evidence seems to suggest that under normal circumstances, with the initial condition we used, the CHR IBVP does indeed remains gradient-stable, as is evident in Fig. 5c. This figure shows that the free-energy is always decreasing, though between time-steps 500 through 1500 (which corresponds to the existence of the travelling wave), the free energy is decreasing very slowly, at a rate of approximately -10^{-4}. Once the concentration wave reaches the boundaries of the domain, the free energy can be decreased significantly (see time step 1750 in Fig. 5c) by raising the concentrations at the boundary, before the simulation ends.

A greater number of time-steps are required when solving the CHR IBVP compared to the CH IBVP, as shown in Fig. 5d. Again, the adaptive time-stepping is extremely useful, especially when the travelling wave is propogating. The size of the time-step taken during this period is around 10^1, before reducing as the concentration wave approaches the boundary.

Fig. 6 shows the convergence of the numerical solution for the CHR IBVP (the relative error is measured in the same manner as the CH IBVP in Section 5.1.1). In Fig. 6a, we can see the rate of convergence is significantly lower when compared to the CH IBVP; the only solution that converges above second order is the equality-constrained case with $p = 5, p_o = 4$. In general, we can see that the equality-constrained method outperforms the weighted least-squares. Fig. 6b shows the value of the first derivative at the left-hand boundary (again the RHS is identical). We can see that all of the solution methods employed satisfy $\partial c/\partial x = 0$ to an order of approximately 1.6.

As we can see, the CHR IBVP is much more difficult to solve than the CH IBVP. Fig. 7 shows the concentration profile at the boundary over time (computed with equality-constrained least-squares with $p = 5, p_o = 4$). We can see that compared to the reference solution, the 150 node case raises the boundary concentration slightly slower (the time span between step 3000 to 3500 is quite small, around $t \approx 20$s). Initially, we were concerned that this was due to the coarse solution incorrectly capturing the travelling wave seen in Fig. 5a. This however, was not the case; the coarse solution matched the reference so-

lution very well. The exponential terms in the flux BC appears to exacerbate any small numerical errors in the estimate of the Laplacian at the boundary, and hence the value of t that corresponds to the concentration wave hitting the boundary. Due to the large miscibility gap between the phases (i.e., the concentration difference between the phases), this causes large relative errors, as shown in Fig. 6a. As such, we note that high order approximations should be used near the boundaries whenever possible. We found that computing the Laplacians at the boundary with $p = 5, p_o = 4$ while using $p = 4, p_o = 3$ for interior faces performed very similarly when compared to using the higher-order approximation throughout the domain.

We should also note that in Fig. 6, for simulations that use greater than approximately 300 nodes with the equality-constrained least-squares, and for all simulations using weighted least-squares, the recoverable error condition for IDA had to be disabled to ensure convergence. This results in a small number of time-steps violating gradient-stability. Fig. 8 shows an example of this behaviour, where 79 time-steps violate strict gradient-stability and these have been highlighted. The values of $\mathrm{d}\tilde{G}/\mathrm{d}t$ at these time steps are in fact quite small, the largest that occurs is $\mathrm{d}\tilde{G}/\mathrm{d}t = 1.11$. Nonetheless, it appears that the best method to solve the CHR problem is to use a high-order approximation at the boundaries, coupled with a small to moderate number of node points to ensure an accurate, strictly gradient-stable solution. As noted above, there is no guarantee that the solution to the CHR IBVP will remain gradient-stable. The dependence of gradient-stability on mesh spacing and solution methodology (along with the resulting small violation) shown in Fig. 8, however, seem to suggest that numerical errors inherent in the discretisation are causing this behaviour, rather than the intrinsic behaviour of the CHR IBVP causing a large, positive increase in $\mathrm{d}\tilde{G}/\mathrm{d}t$, given our initial condition.

5.2 Two-dimensional results

We were unable to compute a "reference" solution for the two-dimensional CH and CHR IBVPs, due to the substantial increase in runtime given the large number of nodes and small tolerances required. As such, the following section does not include plots that show evidence of the convergence of the numerical methods in two-dimensions. We did, however, compute the values of the derivatives $\hat{\mathbf{n}} \cdot (\tilde{\mathbf{K}} \nabla c)$ and $\hat{\mathbf{n}} \cdot (\nabla (\nabla \cdot (\tilde{\mathbf{K}} \nabla c)))$ at $x = -10$ across y (which are identical to the values on other boundaries given the symmetry in our domain and IC), in order to verify our solutions satisfy the BCs. The maximum absolute value of these derivatives are tabulated in Table 1. Also, given the results in the previous section, all the solutions shown in this section were computed using $p = 5$ and $p_o = 4$.

5.2.1 CH IBVP

Fig. 9 and 10 show the results from solving the CH IBVP in two dimensions, with 150×150 node points using the equality-constrained least-squares method. There is, however, one key difference in the solution methodology when compared with Section 5.1. When solving the least-squares problem described in Section 4 with the equality-constrained method, the number of boundary equations we apply, r_t, is greater than $m(p)$. This is because we include both the

No. of Nodes	$\|\|h_{\max}\|\|$	$\hat{\mathbf{n}} \cdot (\tilde{\mathbf{K}} \nabla c)$	$\hat{\mathbf{n}} \cdot (\nabla(\nabla \cdot (\tilde{\mathbf{K}} \nabla c)))$	$\hat{\mathbf{n}} \cdot (\tilde{\mathbf{K}} \nabla c)$	$\hat{\mathbf{n}} \cdot (\nabla(\nabla \cdot (\tilde{\mathbf{K}} \nabla c)))$
CH IBVP		Weighted		Equality	
75×75	0.8653	1.43×10^{-5}	1.31×10^{-3}	1.46×10^{-5}	1.25×10^{-3}
100×100	0.6468	5.92×10^{-6}	8.97×10^{-4}	5.98×10^{-6}	1.2×10^{-3}
125×125	0.5164	3.15×10^{-6}	7.53×10^{-4}	3.21×10^{-6}	1.6×10^{-3}
150×150	0.4297	1.83×10^{-6}	6.22×10^{-4}	2.42×10^{-6}	2.0×10^{-3}
Order		2.92	1.04	2.62	-0.69
CHR IBVP		Weighted		Equality	
100×100	0.6468	6.6×10^{-2}	-	7.9×10^{-2}	-
125×125	0.5164	4.5×10^{-2}	-	5.4×10^{-2}	-
150×150	0.4297	2.9×10^{-2}	-	3.9×10^{-2}	-
Order		2	-	1.73	-

Table 1: Maximum absolute value of the derivatives on the $x = -10$ boundary across y for the CH and CHR IBVPs in two-dimensions, with $p = 5$ and $p_o = 4$. Runtime constrains meant we could not run enough test cases to justify a plot like Fig. 6b.

VBC (9) and (10) in the least-squares problem for the CH IBVP. We cannot, therefore, use the nullspace method to solve the constrained system (36), as it involves partitioning the QR factorisation of $\mathbf{WH}$ into two matrices of size $r_t \times m(p)$ and $m(p) - r_t \times m(p)$. As such, when solving the CH IBVP in two-dimensions with the equality-constrained method, we only constrain the least-squares system to satisfy the VBC (9), rather than both (9) and (10). This problem does not occur when using the weighted least-squares method, and there was no visible difference in the solutions when comparing this modified equality-constrained method with the weighted least-squares method.

Fig. 9 shows that the solution behaves in a similar manner to the one-dimensional solution shown in Fig. 3a. The symmetric initial condition again evolves to a steady state at $t = 180$s where the material has phase separated as much as possible. The numerical solution conserves mass to near machine precision and remains strictly gradient stable, as shown in Fig. 10a and 10b. Fig. 10c plots the size of the time-step taken with time and we can see that the adaptive time-stepping scheme increases the size of the step throughout the simulation. We also note that it takes approximately 50 more time steps than the one-dimensional simulation (see Fig. 3d) in order to reach a steady state.

In terms of satisfying the BCs, Table 1 shows that for the CH IBVP solved using the weighted least-squares method, (9) is satisfied to approximately third order. The condition (10) however, is only satisfied to first order. The equality-constrained method performs even worse, with the VBC (9) being satisfied to below third order and the maximum absolute value of the derivatives in (10) increase with decreasing grid spacing. This is due to our inability to include (10) in the least-squares system when using the equality-constrained method, as described above. The solution produced does not visibly change, but these

results suggest that the weighted least-squares method should be used to solve the CH IBVP in two-dimensions to ensure both BCs are satisfied as the grid spacing decreases.

5.2.2 CHR IBVP

The solution to the CHR IBVP is shown at several time points in Fig. 11, with associated statistics shown in Fig. 12. Like the one-dimensional solution shown in Fig. 5a, the solution profile phase-separates as much as possible, up to $t = 807.7$s. We can see in Fig. 12a that this process is associated with a loss of mass ($t = 807.7$s roughly corresponds to time step 500). Mass then begins to enter the system, and a travelling wave spreads from the circular phase-separated region in the centre of the domain, towards the boundaries. This travelling wave propagates from $t = 807.7$s through to approximately $t = 5660$s, which corresponds to time steps 500 through 1500. The phase-separated region then impacts the boundaries of the domains around $t = 5679.6$s, and the rest of the domain is completely phase-separated a short time later at $t = 5708.1$s. This small time period between $t = 5679.6$s and $t = 5708.1$s is responsible for increasing the total proportion of mass in the system from 0.6 to 1, as shown in Fig. 12c.

The movement of the travelling wave occupies the majority of the simulation time, and the adaptive time-stepping takes large time-steps during this period, as shown in Fig. 12c. Interestingly, Fig. 12b shows us that many of the time-steps taken during the simulation violate strict gradient-stability. Indeed, it appears that the majority of these steps occur when the travelling wave is moving (between steps 500-1500). In the one-dimensional solutions, this is the period when the value of $\mathrm{d}\tilde{G}/\mathrm{d}t$ is extremely small (though still negative). The largest positive value of $\mathrm{d}\tilde{G}/\mathrm{d}t$ that occurs in Fig. 12b during this process is 2.98, which is small given the larger scale of $\mathrm{d}\tilde{G}/\mathrm{d}t$ in the two-dimensional simulation when compared with the one-dimensional (-22500 in Fig. 12b compared with -100 in Fig. 5c). Given the large time-steps taken during the wave propagation and the observed violation of gradient-stability, we restricted the maximum size of the time-step IDA takes to 10^0. Unfortunately this made no difference to the gradient stability of the solution, though it did increase the runtime of the problem considerably. As such, we believe that this behaviour is most likely related to that seen in Fig. 8, where an increase in the number of nodes saw a gradient-stable solution (Fig. 5c) become one that violates gradient stability. In two-dimensions, the least-squares matrix $\mathbf{H}$ is more singular than its one-dimensional equivalent, and any significant decrease in the number of nodes (75×75 for example) in an attempt to ameliorate this behaviour results in the solution not converging. The recoverable error described in Section 5 also had to be disabled to ensure the solutions to the two-dimensional CHR IBVP shown in Fig. 11 converged. Unfortunately, we were unable to find any combination of grid spacing, time-step restrictions, p and p_o value, recoverable error criteria, or least-squares solution methodology to give a perfectly gradient-stable solution to the two-dimensional CHR IBVP.

Table 1 shows that the weighted least-squares method satisfies the VBC (9) to second order when solving the CHR IBVP in two-dimensions, compared with slightly below second order for the equality-constrained method. Given that our discretisation in two-dimensions should be second-order, the weighted least-

squares method should be used to solve the CHR IBVP to ensure the VBC (9) is satisfied to the same order.

6 Conclusions

The least-squares method presented above for solving the CH and CHR IBVPs performs well in both one and two-dimensions. The problem of applying multiple BCs with a fourth-order PDE is handled by incorporating the variational BC into the least-squares system that is used to evaluate information at CV faces.

For the CH IBVP in one-dimension, up to fourth-order convergence is reported, using either a weighted or equality-constrained least-squares method. The solutions produced are gradient-stable and satisfy mass conservation laws to near machine precision. Using no more than 600 node points on a linear grid ensures that the BCs are applied in a high-order manner. The CHR IBVP in one dimension, however, requires the use of high-order Taylor series in the least-squares system, solved using the equality-constrained method to ensure greater than second-order convergence is maintained. Satisfying strict gradient-stability however, requires balancing the conditioning of the least-squares system with the high-order Taylor series required and hence less than 400 node-points with lower-order Taylor approximations should be used.

As might be expected, the two-dimensional IBVPs were more difficult to solve numerically. The solution to the CH IBVP remains gradient stable while satisfying mass conservation laws, whereas no solution to the CHR IBVP could be found that remains strictly gradient-stable. The exponential form of the BCs in the CHR IBVP increase the stiffness of the problem considerably when compared to the CH IBVP, making the gradient-stable solution a challenge to compute. Also, the weighted least-squares methods should be used in two-dimensions, in order to satisfy the BCs sufficiently.

In general, given the lack of an analytic proof that the CHR IBVP should remain gradient-stable, we do not expect our numerical solution to always satisfy this constraint. Our numerical results however, seem to suggest that the initial condition tested in this work could represent an excellent test case for the CHR IBVP, possibly admitting gradient-stable solutions. Indeed, the CHR IBVP represents a test for discretisation schemes that are currently used to solve the CH IBVP, as our scheme conserves and is perfectly gradient-stability in one and two dimensions for the CH IBVP, but fails for the CHR IBVP in two-dimensions. Given the extensive literature on numerical solutions of the CH IBVP and the comparative lack on the CHR IBVP, we believe this offers an exciting opportunity to not only test discretisation methods like this work, but time-stepping schemes on the more difficult CHR IBVP.

Acknowledgements

The authors would like to thank Prof. Ian Turner for helpful discussions and the HPC team at QUT for computing assistance.

References

[1] G. K. Singh, G. Ceder, M. Z. Bazant, Intercalation dynamics in rechargeable battery materials: General theory and phase-transformation waves in $LiFePO_4$, Electrochimica Acta 53 (2008) 7599–7613.

[2] D. Burch, Intercalation Dynamics in Lithium-Ion Batteries, Ph.D. thesis, Massachusetts Institute of Technology, 2009.

[3] J. W. Cahn, J. E. Hilliard, Free energy of a nonuniform system. i. interfacial free energy, The Journal of Chemical Physics 28 (1958) 258.

[4] A. K. Padhi, K. S. Nanjundaswamy, J. B. Goodenough, Phospho-olivines as Positive-Electrode materials for rechargeable lithium batteries, Journal of The Electrochemical Society 144 (1997) 1188–1194.

[5] L. Laffont, C. Delacourt, P. Gibot, M. Wu, P. Kooyman, C. Masquelier, J. Tarascon, Study of the $LiFePO_4/FePO_4$ Two-Phase system by High-Resolution electron energy loss spectroscopy, Chemistry of Materials 18 (2006) 5520–5529.

[6] J. Crank, Free and Moving Boundary Problems, Oxford University Press, Oxford [Oxfordshire], 1984.

[7] V. Srinivasan, J. Newman, Discharge model for the lithium Iron-Phosphate electrode, Journal of The Electrochemical Society 151 (2004) A1517–A1529.

[8] S. Dargaville, T. W. Farrell, Predicting active material utilization in $LiFePO_4$ electrodes using a multiscale mathematical model, Journal of The Electrochemical Society 157 (2010) A830–A840.

[9] S. Nishimura, G. Kobayashi, K. Ohoyama, R. Kanno, M. Yashima, A. Yamada, Experimental visualization of lithium diffusion in Li_xFePO_4, Nature Materials 7 (2008) 707–711.

[10] D. Burch, M. Z. Bazant, Size-dependent spinodal and miscibility gaps for intercalation in nanoparticles, Nano Letters 9 (2009) 3795–3800.

[11] G. Kobayashi, S. Nishimura, M. Park, R. Kanno, M. Yashima, T. Ida, A. Yamada, Isolation of solid solution phases in size-controlled Li_xFePO_4 at room temperature, Advanced Functional Materials 19 (2009) 395–403.

[12] L. Banas, R. Nürnberg, Adaptive finite element methods for Cahn-Hilliard equations, Journal of Computational and Applied Mathematics 218 (2008) 2–11.

[13] S. Zhang, M. Wang, A nonconforming finite element method for the Cahn-Hilliard equation, Journal of Computational Physics 229 (2010) 7361–7372.

[14] S. M. Choo, S. K. Chung, Conservative nonlinear difference scheme for the Cahn-Hilliard equation, Computers & Mathematics with Applications 36 (1998) 31–39.

[15] E. de Mello, O. Teixeira da Silveira Filho, Numerical study of the Cahn-Hilliard equation in one, two and three dimensions, Physica A: Statistical Mechanics and its Applications 347 (2005) 429–443.

[16] L. He, Y. Liu, A class of stable spectral methods for the Cahn-Hilliard equation, Journal of Computational Physics 228 (2009) 5101–5110.

[17] M. Dehghan, D. Mirzaei, A numerical method based on the boundary integral equation and dual reciprocity methods for one-dimensional Cahn-Hilliard equation, Engineering Analysis with Boundary Elements 33 (2009) 522–528.

[18] J. B. Greer, A. L. Bertozzi, G. Sapiro, Fourth order partial differential equations on general geometries, J. Comput. Phys. 216 (2006) 216246. ACM ID: 1150594.

[19] Y. Xia, Y. Xu, C. Shu, Local discontinuous galerkin methods for the Cahn-Hilliard type equations, Journal of Computational Physics 227 (2007) 472–491.

[20] J. Kim, K. Kang, J. Lowengrub, Conservative multigrid methods for Cahn-Hilliard fluids, Journal of Computational Physics 193 (2004) 511–543.

[21] L. Cueto-Felgueroso, J. Peraire, A time-adaptive finite volume method for the Cahn-Hilliard and Kuramoto-Sivashinsky equations, Journal of Computational Physics 227 (2008) 9985–10017.

[22] S. Patankar, Numerical Heat Transfer And Fluid Flow, Taylor & Francis, 1 edition, 1980.

[23] A. Quarteroni, R. Sacco, F. Saleri, Numerical mathematics, Springer, 2000.

[24] L. Cueto-Felgueroso, I. Colomina, F. N. J. Fe, M. Casteleiro, High-order finite volume schemes on unstructured grids using moving least-squares reconstruction. application to shallow water dynamics, International Journal for Numerical Methods in Engineering 65 (2006) 295–331.

[25] I. Prigogine, R. Defay, Chemical thermodynamics, Longmans, Green, 1954.

[26] M. Everdell, Introduction to Chemical Thermodynamics, Hodder & Stoughton Ltd, 1965.

[27] D. Eyre, Unconditionally gradient stable time marching the Cahn-Hilliard equation, 1998. http://www.math.utah.edu/ eyre/research/methods/stable.ps.

[28] J. Newman, K. E. Thomas-Alyea, Electrochemical Systems, Wiley-Interscience, 3 edition, 2004.

[29] J. Pasdunkorale, I. W. Turner, A second order finite volume technique for simulating transport in anisotropic media, International Journal of Numerical Methods for Heat & Fluid Flow 13 (2003) 31–56.

[30] J. A. Belward, I. W. Turner, M. Ilic, On derivative estimation and the solution of least squares problems, Journal of Computational and Applied Mathematics 222 (2008) 511–523.

[31] I. W. Turner, J. A. Belward, M. N. Oqielat, Error bounds for least squares gradient estimates, SIAM Journal on Scientific Computing 32 (2010) 2146.

[32] C. D. Meyer, Matrix Analysis and Applied Linear Algebra, SIAM: Society for Industrial and Applied Mathematics, Har/Cdr edition, 2001.

[33] J. D. Hoffman, Numerical Methods for Engineers and Scientists, Mcgraw-Hill College, 1992.

[34] P. K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, SIAM Journal on Numerical Analysis 21 (1984) 995–1011.

[35] E. Bertolazzi, G. Manzini, A unified treatment of boundary conditions in least-square based finite-volume methods, Computers & Mathematics with Applications 49 (2005) 1755–1765.

[36] G. H. Golub, C. F. V. Loan, Matrix computations, JHU Press, 1996.

[37] A. Bjorck, Numerical methods for least squares problems, SIAM, 1996.

[38] R. Mittal, G. Iaccarino, Immersed boundary methods, Annual Review of Fluid Mechanics 37 (2005) 239–261.

[39] R. Ghias, R. Mittal, H. Dong, A sharp interface immersed boundary method for compressible viscous flows, Journal of Computational Physics 225 (2007) 528–553.

[40] J. R. Pacheco, A. Pacheco-Vega, T. Rodi, R. E. Peck, Numerical simulations of heat transfer and fluid flow problems using an immersed-boundary finite-volume method on nonstaggered grids, Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology 48 (2005) 1.

[41] T. Davis, Multifrontal multithreaded rank-revealing sparse QR factorization, ACM Transactions on Mathematical Software, submitted for publication (-).

[42] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, C. S. Woodward, SUNDIALS: suite of nonlinear and differential/algebraic equation solvers, ACM Transactions on Mathematical Software 31 (2005) 363396. ACM ID: 1089020.

[43] B. P. Vollmayr-Lee, A. D. Rutenberg, Fast and accurate coarsening simulation with an unconditionally stable time step, Physical Review E 68 (2003) 066703.

[44] A.-k. Kassam, Lloyd, N. Trefethen, Fourth-order time stepping for stiff PDEs, SIAM Journal on Scientific Computing 26 (2005) 1214—1233.

[45] Y. He, Y. Liu, T. Tang, On large time-stepping methods for the Cahn-Hilliard equation, Applied Numerical Mathematics 57 (2007) 616–628.

[46] S. M. Choo, S. K. Chung, K. I. Kim, Conservative nonlinear difference scheme for the Cahn-Hilliard equation–II, Computers & Mathematics with Applications 39 (2000) 229–243.

[47] J. M. Hyman, R. J. Knapp, J. C. Scovel, High order finite volume approximations of differential operators on nonuniform grids, Physica D: Nonlinear Phenomena 60 (1992) 112–138.

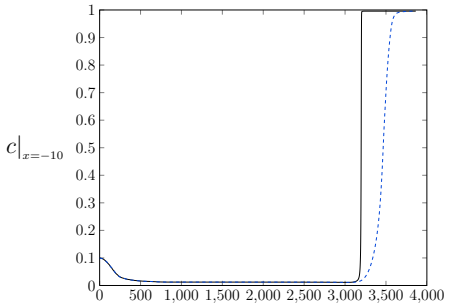


Figure 7: Concentration at $x = -10$ over time. The solid line is from the reference solution (computed with 1000 nodes), the dotted line uses 150 nodes. The reference solution has been linearly interpolated to be at the same time points as the coarser solution.

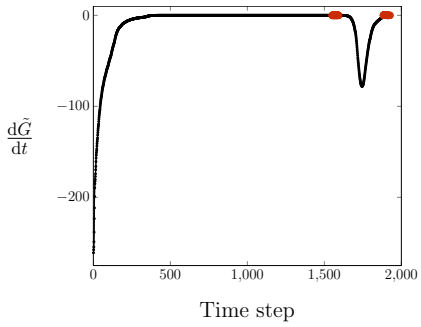
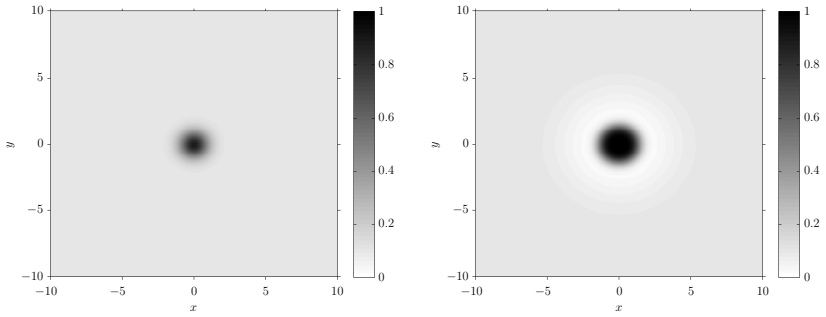


Figure 8: Free energy over time with 400 nodes using equality-constrained least-squares with $p = 5, p_o = 4$ and the recoverable error disabled. There are 79 time steps that violate gradient-stability and these have been highlighted in red.



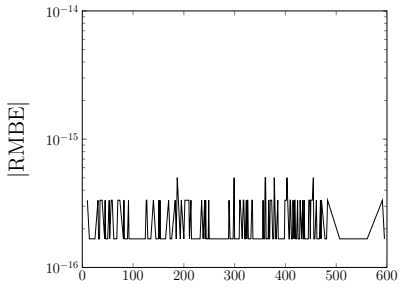
(a) t=0s. Initial condition: $0.1 + 0.8\exp(-(x^2+y^2))$.

(b) t=4.94s

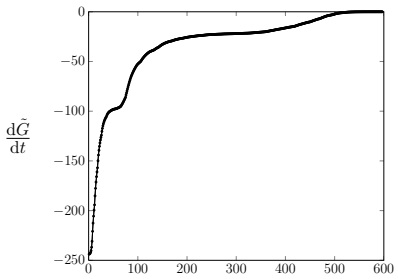
(c) t=66.7s

(d) t=180s

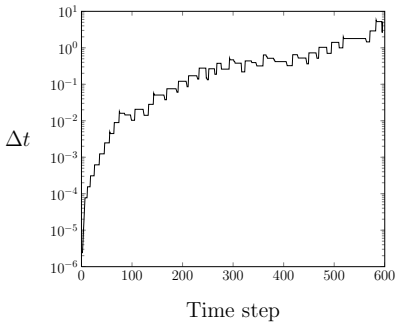
Figure 9: CH IBVP solution computed in two dimensions ($x, y \in [-10, 10]$) with 150×150 nodes using equality-constrained least-squares. Relative and absolute tolerances were set at 1×10^{-6}. $\tilde{\Omega}_{\mathrm{m}} = 5$, $\tilde{\mathbf{K}} = \mathbf{I}$, $\mathbf{D} = \mathbf{I}$, $p = 5$, $p_o = 4$.



(a) Relative mass balance error over time.

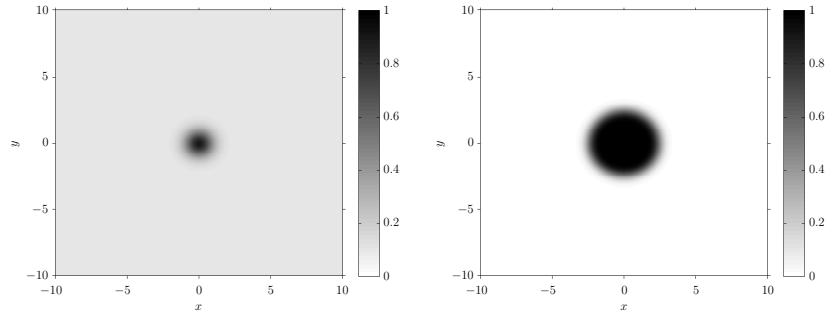


(b) Change in free energy over time.



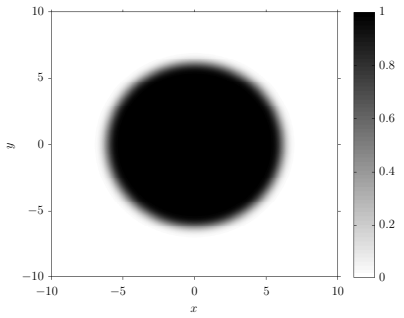
(c) Size of the time steps taken.

Figure 10: CH IBVP solution statistics for the simulation shown in Fig. 9.

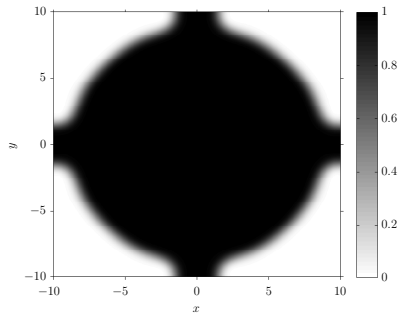


(a) t=0s. Initial condition: $0.1 + 0.8\exp(-(x^2+y^2))$.

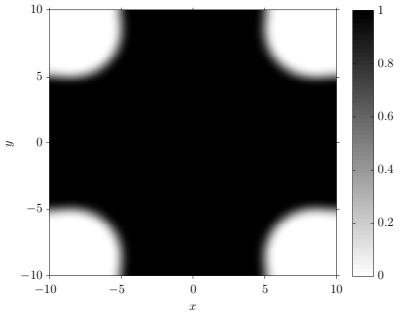
(b) t=807.7s



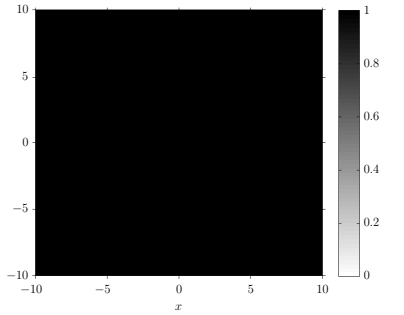
(c) t=4378.5s



(d) t=5679.6s

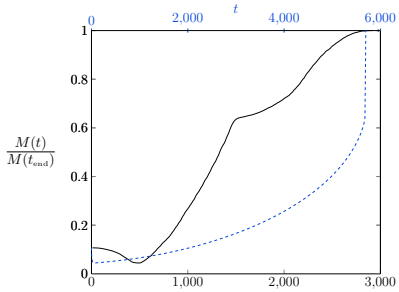


(e) t=5685.5s

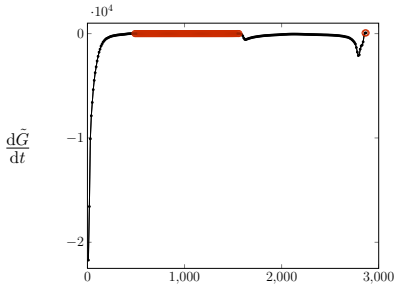


(f) t=5708.1s

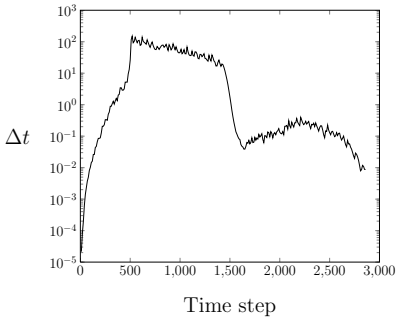
Figure 11: CHR IBVP solution computed in two dimension ($x, y \in [-10, 10]$) with 150 $\times$ 150 nodes using equality-constrained least-squares. Relative and absolute tolerances were set at 1×10^{-6}. $\tilde{\Omega}_{\mathrm{m}} = 5$, $\tilde{\mathbf{K}} = \mathbf{I}$, $\mathbf{D} = \mathbf{I}$, $\rho_{\mathrm{s}} R_{\mathrm{ins}}/\rho = 1$, $\tilde{\mu}_e = 0.5$, $p = 5$, $p_o = 4$.



(a) Proportion of mass in the system over time. The solid line corresponds to the bottom x axis (time step), while the dashed line corresponds to the top x axis (t).



(b) Change in free energy over time. There are 1100 time steps that violate gradient-stability and these have been highlighted in red.



(c) Size of the time steps taken.

Figure 12: CHR IBVP solution statistics for the simulation shown in Fig. 11.